

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-282713

(43) 公開日 平成11年(1999)10月15日

| (51) Int.Cl. ⁸ | 識別記号 | F I |
|---------------------------|-------|-----------------------|
| G 0 6 F 11/22 | 3 4 0 | G 0 6 F 11/22 3 4 0 A |
| 11/28 | | 11/28 A |
| 15/78 | 5 1 0 | 15/78 5 1 0 K |

審査請求 未請求 請求項の数12 F D (全 16 頁)

(21) 出願番号 特願平10-103719

(22) 出願日 平成10年(1998) 3月31日

(71) 出願人 000002369

セイコーエプソン株式会社

東京都新宿区西新宿 2 丁目 4 番 1 号

(72) 発明者 工藤 真

長野県諏訪市大和 3 丁目 3 番 5 号 セイコーエプソン株式会社内

(72) 発明者 土方 陽一

長野県諏訪市大和 3 丁目 3 番 5 号 セイコーエプソン株式会社内

(72) 発明者 宮山 芳幸

長野県諏訪市大和 3 丁目 3 番 5 号 セイコーエプソン株式会社内

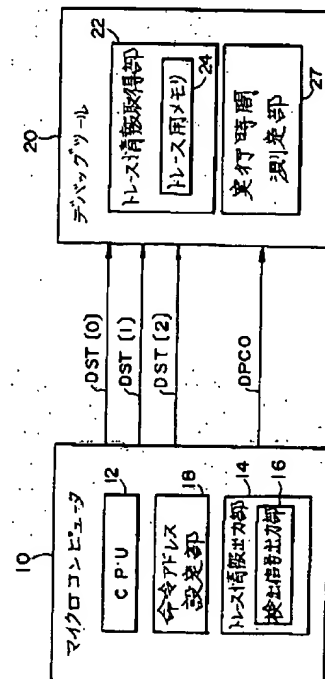
(74) 代理人 弁理士 井上 一 (外 2 名)

(54) 【発明の名称】 マイクロコンピュータ、電子機器及びデバッグシステム

(57) 【要約】

【課題】 少ない端子で量産チップ上でのリアルタイムトレースを実現し、指定された範囲のトレース情報の取得及び実行時間の測定を行うことができるマイクロコンピュータ、これを含む電子機器、及びデバッグシステムを提供する。

【解決手段】 トレース情報出力部 16 は、リアルタイムトレースを実現するためのトレース情報を専用 4 端子に出力する。CPU の命令実行ステータス情報 (DST [2:0]) を 3 端子に出力し、PC 絶対分岐実行が発生すると分岐先の PC 値 (DPCO) をシリアルに 1 端子に出力する。マイクロコンピュータ 10 側は所定の順番でトレース範囲又は実行時間測定範囲のスタートとエンドを示す情報を DST [2] に出力する。デバッグツール 20 側は DST [2] の値に基づきトレース範囲又は実行時間測定範囲のスタートとエンドを判断する。



【特許請求の範囲】

【請求項1】 リアルタイムトレース機能を有するマイクロコンピュータであって、
命令の実行処理を行う中央処理ユニットと、
前記中央処理ユニットで実行されるプログラムの複数の命令の実行を検出するための命令アドレスを設定する命令アドレス設定手段と、

前記複数の命令の実行を検出した場合、検出信号を1本の検出信号出力端子を介して外部に出力する検出信号出力手段とを含むことを特徴とするマイクロコンピュータ。

【請求項2】 請求項2において、
前記複数の命令は第一の命令及び第二の命令を含み、
前記検出信号出力手段は、
前記第一の命令が実行されて検出信号が出力された後に、続けて第一の命令が実行された場合には検出信号の出力を省略し、第二の命令が実行された場合にのみ検出信号を出力することを特徴とするマイクロコンピュータ。

【請求項3】 請求項1又は2のいずれかにおいて、
リアルタイムトレースに必要な前記中央処理ユニットの命令実行のステータス情報をステータス情報出力端子に出力するステータス情報出力手段を含み、前記ステータス情報出力用端子の1つを前記検出信号出力端子と共用することを特徴とするマイクロコンピュータ。

【請求項4】 請求項1から3のいずれかにおいて、
前記命令アドレス設定手段が設定するアドレスは、プログラムのトレース範囲を特定するためのスタートアドレスとエンドアドレスを含むことを特徴とするマイクロコンピュータ。

【請求項5】 請求項3又は4のいずれかにおいて、
ステータス情報出力手段は、
トレースに必要なステータス情報として、通常命令実行、相対分岐命令実行、絶対分岐命令実行、通常命令実行時に前記命令アドレスにマッチ、相対分岐命令実行時に前記命令アドレスにマッチ、絶対分岐命令実行時に前記命令アドレスにマッチの状態を表すことのできる情報を含むことを特徴とするマイクロコンピュータ。

【請求項6】 請求項1から5のいずれかにおいて、
前記命令アドレス設定手段が設定するアドレスは、プログラムの実行時間測定範囲を特定するためのスタートアドレスとエンドアドレスを含むことを特徴とするマイクロコンピュータ。

【請求項7】 請求項1～6のいずれかにおいて、
前記命令アドレス設定手段は、
複数の命令の実行を検出するための命令アドレス情報を保持する複数のレジスタを含み、
前記検出信号出力手段は、
前記中央処理ユニットの命令実行のために先読みされた命令アドレスと前記複数のレジスタに保持されたアドレ

スとの比較結果を前記先読みされた命令アドレスの命令が実行されるまで保持し、前記先読みされた命令アドレスの命令が実行された場合に前記保持された比較結果に基づき検出信号を出力する手段と、前記実行された命令が分岐命令であった場合には、当該分岐命令実行以前に先読みされた命令との比較結果を無効にする手段と、を含むことを特徴とするマイクロコンピュータ。

【請求項8】 請求項7において、
前記レジスタは、
プログラムの所与の範囲を特定するためのスタートアドレス及びエンドアドレスを保持する少なくとも2つのレジスタを含み、
検出信号出力手段は、
前記中央処理ユニットが命令実行のために先読みした命令アドレスと前記スタートアドレス及びエンドアドレスとの比較を行う第一の比較手段及び第二の比較手段と、
前記第一の比較手段及び第二の比較手段の比較結果を複数回分保持できる比較結果保持部と、
前記比較結果保持部に保持された前記比較結果を、前記先読みした命令が実行されたタイミングで読み出す手段と、
実行された命令が分岐命令であった場合、前記比較結果保持部に保持されている比較結果をリセットする手段と、
前記比較結果保持部から読み出された比較結果に基づき、所定の順番で前記第一のレジスタ及び第二のレジスタに保持された実行アドレスが実行された場合に検出信号を出力する論理回路とを含むことを特徴とするマイクロコンピュータ。

【請求項9】 請求項1乃至6のいずれかのマイクロコンピュータと、
前記マイクロコンピュータの処理対象となるデータの入力源と、
前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする電子機器。

【請求項10】 マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、
命令の実行処理を行う中央処理ユニットと、
前記中央処理ユニットで実行されるプログラムの複数の命令の実行を検出するための命令アドレスを設定する命令アドレス設定手段と、
前記複数の命令の実行を検出した場合、検出信号を1本の検出信号出力端子を介して外部に出力する検出信号出力手段とを含むマイクロコンピュータと、
前記マイクロコンピュータから前記検出信号を受信し、
前記検出信号に基づきトレース情報の取得開始及び終了の制御を行いトレース情報を取得するトレース情報取得手段とを含むことを特徴とするデバッグシステム。

【請求項11】 マイクロコンピュータを含むターゲッ

トシステムのためのデバッグシステムであって、
命令の実行処理を行う中央処理ユニットと、
前記中央処理ユニットで実行されるプログラムの複数の
命令の実行を検出するための命令アドレスを設定する命
令アドレス設定手段と、

前記複数の命令の実行を検出した場合、検出信号を1本
の検出信号出力端子を介して外部に出力する検出信号出
力手段とを含むマイクロコンピュータと、

前記マイクロコンピュータから前記検出信号を受信し、
前記検出信号に基づきプログラム実行時間の測定の開始
及び終了の制御を行い実行時間を測定する実行時間測定
手段とを含むことを特徴とするデバッグシステム。

【請求項12】 請求項10又は11において、
前記複数の命令は第一の命令及び第二の命令を含み、
前記検出信号出力手段は、

前記第一の命令が実行されて検出信号が出力された後
に、続けて第一の命令が実行された場合には検出信号の
出力を省略し、第二の命令が実行された場合にのみ検出
信号を出力することを特徴とするデバッグシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マイクロコンピ
ュータ、マイクロコンピュータを含む電子機器、及びデバ
ッグシステムに関する。

【0002】

【背景技術及び発明が解決しようとする課題】近年、ゲ
ーム装置、カーナビゲーションシステム、プリンタ、携
帯情報端末などの電子機器に組み込まれ、高度な情報処
理を実現できるマイクロコンピュータに対する需要が高
まっている。このような組み込み型のマイクロコンピ
ュータは、通常、ターゲットシステムと呼ばれるユーザボ
ードに実装される。このターゲットシステムを動作させ
るソフトウェアの開発を支援するためにICE(In-Cir
cuit Emulator)と呼ばれるソフトウェア開発支援ツ
ールが広く使用されている。

【0003】さて、このようなICEとしては、従来、
図1に示すようなCPU置き換え型と呼ばれるICEが
主流を占めていた。このCPU置き換え型ICEでは、
デバッグ時にターゲットシステム300からマイクロコ
ンピュータ302を取り外し、その代わりにデバッグツ
ール304のプロープ306を接続する。そして、この
デバッグツール304に、取り外したマイクロコンピ
ュータ302の動作をエミュレートさせる。また、このデ
バッグツール304に、トレース情報の取得やデバッグ
のために必要な種々の処理を行わせる。

【0004】しかしながら、このCPU置き換え型ICE
では、マイクロコンピュータ302の内部動作周波数
が上がるとプロープ306やトレース情報を格納するバッ
ファで生じる信号の遅延によりリアルタイムトレース
が困難になる。

【0005】またトレース情報を無制限にトレース用の
バッファに格納しようとするトレース用のバッファが
すぐにあふれてしまうので、デバッグに必要な部分のト
レース情報が取得できない場合もある。このような場
合、トレース範囲を指定してトレース情報をとることが
できると、大変便利である。しかしCPUの内部動作周
波数が上がると、CPU置き換え型ICEのように外部
回路でトレースの範囲指定を実現することは困難とな
る。

【0006】このような問題を解決するために量産チッ
プ上でリアルタイムトレースを可能にするシステムの開
発が行われている。このとき、トレース情報としてアド
レスバス等の情報をリアルタイムにトレースバッファに
格納しようとする数多くの専用端子が必要となる。し
かし、係る端子はデバッグ時のみ必要な物で、エンドユ
ーザーにとっては不要なものであるから、より少ないほ
うが好ましい。

【0007】また、量産チップ上でリアルタイムトレ
ースを実現する際にも、指定した範囲のトレース情報が確
実に取得できる機能があると大変便利である。

【0008】本発明は、以上のような技術的課題に鑑み
てなされたものであり、その目的とするところは、少な
い端子で量産チップ上でリアルタイムトレースを実現
し、指定された範囲のトレース情報の取得及び実行時間
の測定を行うことができるマイクロコンピュータ、これ
を含む電子機器、及びデバッグシステムを提供すること
にある。

【0009】

【課題を解決するための手段】上記課題を解決するた
めに本発明は、リアルタイムトレース機能を有するマイク
ロコンピュータであって、命令の実行処理を行う中央処
理ユニットと、前記中央処理ユニットで実行されるプロ
グラムの複数の命令の実行を検出するための命令アドレ
スを設定する命令アドレス設定手段と、前記複数の命令
の実行を検出した場合、検出信号を1本の検出信号出力
端子を介して外部に出力する検出信号出力手段とを含む
ことを特徴とする。

【0010】本発明によれば1本の端子で、複数の命令
が実行された旨の情報を取得することができる。従って
リアルタイムトレースにおいて、ユーザーの使用できる
端子数をほとんど減らすことなく複数の命令の実行の発
生を検出することができ、デバッグ効率の向上を図るこ
とができる。

【0011】また本発明は、前記複数の命令は第一の命
令及び第二の命令を含み、前記検出信号出力手段は、前
記第一の命令が実行されて検出信号が出力された後に、
続けて第一の命令が実行された場合には検出信号の出力
を省略し、第二の命令が実行された場合にのみ検出信号
を出力することを特徴とする。

【0012】1本の端子で複数の命令の実行情報を取得

する場合、どの命令が実行されたのか外部から判断するのは困難である。

【0013】しかし本発明によれば、当該検出信号が前記第一の及び第二の命令が予定されていた順序と異なった順序で実行された場合には、検出信号が出力されない。このため、外部には常に予定された順序で第一の及び第二の命令の実行を示す検出信号が出力される。

【0014】従って本発明によれば、複数の命令が予定されていた順序と異なった順序で実行された場合にも1本の端子で、どの命令が実行されたか判別可能な情報を取得することができる。

【0015】また本発明は、リアルタイムトレースに必要な前記中央処理ユニットの命令実行のステータス情報をステータス情報出力端子に出力するステータス情報出力手段を含み、前記ステータス情報出力用端子の1つを前記検出信号出力端子と共用することを特徴とする。

【0016】本発明によれば、検出信号出力端子を別に設けなくてもよいので、デバッグ用端子の更なる節約なる。

【0017】また本発明は、前記命令アドレス設定手段が設定するアドレスは、プログラムのトレース範囲を特定するためのスタートアドレスとエンドアドレスを含むことを特徴とする。

【0018】本発明によれば、トレース範囲のスタート及びエンドを検出する情報を1本の端子で出力することができる。このため、少ない端子で量産チップ上でのリアルタイムトレースを実現し、指定された範囲のトレース情報を取得可能なマイクロコンピュータを提供することができる。

【0019】また本発明は、ステータス情報出力手段が、トレースに必要なステータス情報として、通常命令実行、相対分岐命令実行、絶対分岐命令実行、通常命令実行時に前記命令アドレスにマッチ、相対分岐命令実行時に前記命令アドレスにマッチ、絶対分岐命令実行時に前記命令アドレスにマッチの状態を表すことのできる情報を含むことを特徴とする。

【0020】ここにおいて、PC相対分岐命令とはプログラム中に明示的に記述されたアドレスに分岐する命令であり、ソースコードから分岐先を判断可能な命令である。またPC絶対分岐命令とはプログラムの実行中にレジスタの値によって分岐先のアドレスが決まる命令であり、ソースコードからは分岐先を判断することができない命令である。

【0021】本発明によれば、6種類のステータス情報を含み、係るステータス情報は3本の端子で実現することができる。このためわずか3本という少ない端子で、指定された範囲のリアルタイムトレースを行うことができる。

【0022】また本発明は、前記命令アドレス設定手段が設定するアドレスは、プログラムの実行時間測定範囲

を特定するためのスタートアドレスとエンドアドレスを含むことを特徴とする。

【0023】本発明によれば、実行時間測定範囲のスタート及びエンドを検出する情報を1本の端子で出力することができる。このため、少ない端子で、指定された範囲の実行時間測定の開始及び終了のタイミングを判断するための情報を取得可能なマイクロコンピュータを提供することができる。

【0024】また本発明は、前記命令アドレス設定手段が、複数の命令の実行を検出するための命令アドレス情報を保持する複数のレジスタを含み、前記検出信号出力手段は、前記中央処理ユニットの命令実行のために先読みされた命令アドレスと前記複数のレジスタに保持されたアドレスとの比較結果を前記先読みされた命令アドレスの命令が実行されるまで保持し、前記先読みされた命令アドレスの命令が実行された場合に前記保持された比較結果に基づき検出信号を出力する手段と、前記実行された命令が分岐命令であった場合には、当該分岐命令実行以前に先読みされた命令との比較結果を無効にする手段と、含むことを特徴とする。

【0025】ここにおいて比較結果を無効にするとは、保持エリアを初期化する場合や書き込み及び読み出し用のポインタをリセットする場合を含む。

【0026】本発明では、複数の命令の実行の検出を先読みされた命令に基づいて行い、検出信号の出力は実際に実行されてから行う。このようにすることでパイプライン処理を行うマイクロコンピュータでも、最適な検出を行うことができる。

【0027】また本発明は、前記レジスタが、プログラムの所与の範囲を特定するためのスタートアドレス及びエンドアドレスを保持する少なくとも2つのレジスタを含み、検出信号出力手段は、前記中央処理ユニットが命令実行のために先読みした命令アドレスと前記スタートアドレス及びエンドアドレスとの比較を行う第一の比較手段及び第二の比較手段と、前記第一の比較手段及び第二の比較手段の比較結果を複数回分保持できる比較結果保持部と、前記比較結果保持部に保持された前記比較結果を、前記先読みした命令が実行されたタイミングで読み出す手段と、実行された命令が分岐命令であった場合、前記比較結果保持部に保持されている比較結果をリセットする手段と、前記比較結果保持部から読み出された比較結果に基づき、所定の順番で前記第一のレジスタ及び第二のレジスタに保持された実行アドレスが実行された場合に検出信号を出力する論理回路とを含むことを特徴とする。

【0028】パイプライン処理においては、命令の取得と実行にタイムラグがあるため実行時には、信号線から実行アドレス情報を取得することが困難である。本発明によれば命令実行のために先読みされた段階での比較結果を実行された段階で取得することができる。このため

パイプライン処理に対応した信号処理を行うことができる。

【0029】なお分岐命令が実行された場合には、比較結果保持部に保持されている分岐命令以前の先読みされた命令との比較結果をリセットすることで分岐命令発生により先読みされた命令が実行されない場合にも正確な判断を行うことができる。

【0030】また本発明に係る電子機器は、上記いずれかのマイクロコンピュータと、前記マイクロコンピュータの処理対象となるデータの入力源と、前記マイクロコンピュータにより処理されたデータを出力するための出力装置とを含むことを特徴とする。

【0031】このようにすれば電子機器を動作させるプログラムなどのデバッグ作業の効率化を図れるようになり電子機器の開発期間の短縮化及び低コスト化を図れるようになる。

【0032】また本発明はマイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、命令の実行処理を行う中央処理ユニットと、前記中央処理ユニットで実行されるプログラムの複数の命令の実行を検出するための命令アドレスを設定する命令アドレス設定手段と、前記複数の命令の実行を検出した場合、検出信号を1本の検出信号出力端子を介して外部に出力する検出信号出力手段とを含むマイクロコンピュータと、前記マイクロコンピュータから前記検出信号を受信し、前記検出信号に基づきトレース情報の取得開始及び終了の制御を行いトレース情報を取得するトレース情報取得手段とを含むことを特徴とする。

【0033】本発明によれば少ない端子で指定された範囲のデバッグ情報を取得できるデバッグシステムを提供することができる。

【0034】また本発明は、マイクロコンピュータを含むターゲットシステムのためのデバッグシステムであって、命令の実行処理を行う中央処理ユニットと、前記中央処理ユニットで実行されるプログラムの複数の命令の実行を検出するための命令アドレスを設定する命令アドレス設定手段と、前記複数の命令の実行を検出した場合、検出信号を1本の検出信号出力端子を介して外部に出力する検出信号出力手段とを含むマイクロコンピュータと、前記マイクロコンピュータから前記検出信号を受信し、前記検出信号に基づきプログラム実行時間の測定の開始及び終了の制御を行い実行時間を測定する実行時間測定手段とを含むことを特徴とする。

【0035】本発明によれば少ない端子で指定された範囲の実行時間を取得できるデバッグシステムを提供することができる。

【0036】また本発明のデバッグシステムは、前記複数の命令は第一の命令及び第二の命令を含み、前記検出信号出力手段は、前記第一の命令が実行されて検出信号が出力された後に、続けて第一の命令が実行された場合

には検出信号の出力を省略し、第二の命令が実行された場合にのみ検出信号を出力することを特徴とする。

【0037】本発明によれば、複数の命令が予定されていた順序と異なった順序で実行された場合にも、1本の端子でデバッグどの命令が実行されたか判別可能なデバッグシステムを提供することができる。

【0038】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0039】1. 本実施形態の特徴

まず本実施形態の構成について図2を用いて説明する。

【0040】図2に示すように、本実施形態では、マイクロコンピュータ10が、CPU（中央処理ユニット）12及び命令アドレス設定部18、トレース情報出力部14を含む。トレース情報出力部14は検出信号出力部16を含む。

【0041】また、マイクロコンピュータ10の外部のデバッグツール20はトレース情報取得部22、実行時間測定部27を含んでいる。

【0042】ここでトレース情報出力部16は、リアルタイムトレースを実現するためのトレース情報を専用4端子に出力する。具体的には毎クロック毎にCPUの命令実行ステータス情報（DST[2:0]）を3端子に出力し、ステータス情報出力手段として機能する。またPC絶対分岐実行が発生すると続く27クロックサイクルで分岐先のPC値（DPCO）をシリアルに1端子に出力する。

【0043】図3はDST[2:0]の出力値とCPUの命令実行状態の関係を表した図である。ここにおいてPC相対分岐命令とはプログラム中に明示的に記述されたアドレスに分岐する命令であり、ソースコードから分岐先を判断可能である。またPC絶対分岐命令とはプログラムの実行中にレジスタの値によって分岐先のアドレスが決まる命令であり、ソースコードからは分岐先を判断することができない。

【0044】従って本実施の形態ではPC絶対分岐命令が発生すると分岐先のPC値（DPCO）を出力することでトレースを可能としている。

【0045】即ちマイクロプロセッサが命令をアドレスの順番に実行している時（DST[2:0]の出力値が000又は100である場合）には、プログラムカウンタがどこまで進んだか分かるのでプログラムのソースコードからトレースが可能である。また、マイクロプロセッサがPC相対分岐命令を実行した時（DST[2:0]の出力値が001又は101である場合）にも、プログラムのソースコードから分岐先が分かるのでトレースが可能である。

【0046】しかし、マイクロプロセッサがPC絶対分岐命令を実行した時（DST[2:0]の出力値が010又は110である場合）には、プログラムのソースコ

ードから分岐先が分からないため、DST[2:0]のみではトレースができない。そこで、係る場合に分岐先のPC値(DPCO)を出力することでトレースを可能としているのである。

【0047】ここで本実施の形態の特徴的な事項である1本の端子でトレース範囲を特定する点について説明する。通常範囲指定を行わない場合にはトレース情報(DST[2:0]とDPCO)はトレース用メモリ24順次書き込まれ満杯になると古い情報を消して新しい情報をオーバーライトするよう構成されている。しかしトレース用メモリはすぐにあふれてしまうのでユーザーが必要としている部分のトレース情報が取得できない場合が多い。そこで本実施の形態では、前記DST[2:0]の値から、トレース範囲の取得開始位置及び終了位置を検出できるような構成としている。

【0048】以下前記DST[2:0]の値から、トレース範囲の取得開始位置及び終了位置を検出する構成について説明する。

【0049】図3の312はブレーク空間、即ちデバッグモードでは出力されるDSTが111であることを示している。デバッグモードの間はユーザープログラムは実行されないため、ユーザープログラム実行時にDST[2]が1となるのは310の3つの場合のみである。即ち、通常命令又はPC相対命令又はPC絶対命令のいずれかが命令ブレークにマッチしている状態である。ここにおいて命令ブレークとは、命令アドレス設定手段に設定されている命令アドレスの意味である。ここでは、プログラムのトレース範囲を特定するためのスタートアドレス及びエンドアドレスが命令ブレークとして命令アドレス設定手段に設定されている。従ってマイクロコンピュータから出力されるDST[2]が1であれば、デバッグツール20側は、マイクロコンピュータ10でスタートアドレス又はエンドアドレスの命令が実行されたことを検出することができる。

【0050】なお、本実施の形態では命令ブレークは実際にはブレークせず実行が継続されるが、実際にブレークを起こすチップ内のブレーク機能を兼用で使用している。

【0051】次に、DST[2]が1であるとき、スタートアドレスの命令が実行されたのかエンドアドレスの命令が実行されたのかを判別する構成について説明する。

【0052】図4(A)(B)(C)は、本実施の形態における命令ブレークにマッチした場合のDST[2:0]の出力を説明するための図である。

【0053】命令ブレーク#2にマッチとは、スタートアドレスの命令が実行された場合を示し、命令ブレーク#1にマッチとは、エンドアドレスの命令が実行された場合を示す。即ち命令ブレーク#2から命令ブレーク#1の間がトレース情報を取得する範囲となる。

【0054】図4(A)の350、352、354、356に示すように命令ブレーク#2及び命令ブレーク#1がこの順で繰り返し発生する場合には命令ブレーク#2にマッチでトレース情報の取得を開始し、命令ブレーク#1にマッチでトレース情報の取得を終了する処理を繰り返せばよい。

【0055】しかし例えばプログラムの分岐等により命令ブレーク#2及び命令ブレーク#1に所定の順序でマッチしない場合がある。このような場合は、最初に命令ブレーク#2にマッチしたらマッチしたことを示すステータス(以下マッチステータスという。なお、本実施の形態ではDST[2]=1)を出力し、次に命令ブレーク#1にマッチしたところでマッチステータスを出力する。この間、命令ブレーク#2にマッチしてもマッチステータスは出力しない。そしてまた、命令ブレーク#2にマッチしたらマッチステータスを出力し、次に命令ブレーク#1にマッチしたところでマッチステータスを出力する動作を繰り返す。

【0056】図4(B)(C)は、命令ブレーク#1及び命令ブレーク#2のマッチが不規則に行われる例を説明するための図である。図4(C)に示すように318から320まで通常命令を順次実行していたプログラムは320で分岐命令により322に分岐する。322から326へ実行過程で命令ブレーク#1にマッチするが(324)、図4(B)の(注1)に示すようにDST[2:0]からマッチステータスは出力されない。最初は命令ブレーク#2にマッチしたときのみマッチステータスを出力する仕様だからである。

【0057】326から328に分岐したプログラムは、328から332への実行過程で命令ブレーク#2にマッチし(330)、図4(B)の(注3)に示すようにDST[2:0]からマッチ示すステータスを出力する。

【0058】332から334に分岐したプログラムは、334から順次実行する過程で命令ブレーク#2及び命令ブレーク#1にマッチする(336、338)。しかし、2番目は命令ブレーク#1にマッチしたときにマッチステータスを出力するよう構成されているので、図4(B)に示すように336ではマッチステータスは出力されず(注2)、338でマッチステータスが出力される(注4)。従って、330から338までの実線で示された範囲がトレース範囲となる(図4(C)参照)。

【0059】図5は、DST[2]とトレース範囲の関係を示す図である。同図に示すように、DST[2]奇数番目のマッチステータス出力時(360、364)には命令ブレーク#2にマッチした状態にあり、偶数番目のマッチステータス出力時(362、366)には命令ブレーク#1にマッチした状態である。従ってデバッグツール20のトレース情報取得部22は、360から3

62までのトレース範囲1(370)のトレース情報をトレース用メモリ24に格納し、次に364から366までのトレース範囲2(372)のトレース情報をトレース用メモリ24に格納する。

【0060】このように、パルスの順番とトレース範囲のスタートとエンドとの関係を定めておくと、デバッグツール側は奇数番目のパルスであるか偶数番目のパルスであるかによりトレースのスタート、ストップを判断することができる。従って1つの端子のみでトレース範囲の指定を行うことができる。

【0061】このトレース範囲の指定機能は、実行時間を測定するための実行時間測定範囲の特定にも使うことができる。即ち、トレース範囲と同様、実行時間の測定を行いたい範囲のスタートアドレス及びエンドアドレスを命令ブレイクとしてセットして、前記デバッグツール20の実行時間測定部27で所望の範囲の実行時間の測定を行うことができる。

【0062】このようにすることで、少ない端子で量産チップ上でリアルタイムトレースを実現し、指定された範囲のトレース情報の取得及び実行時間の測定を行うことができる。

【0063】なお、実行時間の測定はトレースとは独立して行うことも可能である。即ちトレース情報の取得は行わず、実行時間の測定のみ行うことも可能である。

【0064】2. 詳細な構成例

図6に本実施形態のマイクロコンピュータ及びデバッグシステムの詳細な構成例を示す。図6に示すように、マイクロコンピュータ10は、CPU12、BCU(バス制御ユニット)26、内部メモリ28、クロック生成部30、オンチップモニタ部40、トレース情報出力部14を含む。

【0065】ここでCPU12は、種々の命令の実行処理を行うものであり、内部レジスタ13を含む。内部レジスタ13は、汎用レジスタであるR0～R15や、特殊レジスタであるSP(スタックポインタレジスタ)、AHR(積和結果データの上位レジスタ)、ALR(積和結果データの低位レジスタ)などを含む。

【0066】BCU26はバスを制御するものである。例えば、CPU12に接続されるハーバードアーキテクチャのバス31や、内部メモリ28に接続されるバス32や、外部メモリ36に接続される外部バス33や、トレース情報出力部14などに接続される内部バス34の制御を行う。

【0067】またクロック生成部30は、マイクロコンピュータ10内で使用される各種のクロックを生成するものである。クロック生成部30はCLKを介して外部のデバッグツール60にもクロックを供給している。

【0068】オンチップモニタ部40は、オンチップモニタROM42、オンチップモニタRAM44、制御レジスタ46、SIO48を含む。

【0069】ここで、オンチップモニタROM42には、オンチップモニタプログラムが格納される。本実施形態では、このオンチップモニタプログラムは、GO、リード、ライトなどのシンプルでプリミティブなコマンドの処理のみを行うようになっている。このため、オンチップモニタROM42のメモリ容量を例えば256バイト程度に抑えることができ、オンチップデバッグ機能を持たせながらマイクロコンピュータ10を小規模化できるようになる。

10 【0070】オンチップモニタRAM44には、デバッグモードへの移行時に、CPU12の内部レジスタ13の内容が退避される。これにより、デバッグモードの終了後にユーザプログラムの実行を適正に再スタートできるようになる。また内部レジスタ13の内容のリード等を、オンチップモニタプログラムが持つプリミティブなリードコマンド等で実現できるようになる。

【0071】制御レジスタ46は、各種のデバッグ処理を制御するためのレジスタであり、各種イネーブルビットや命令ブレイクアドレスレジスタなどを有する。オンチップモニタプログラムにより動作するCPU12が制御レジスタ46の各ビットにデータをライトしたり、各ビットのデータをリードすることで、各種のデバッグ処理が実現される。なお、トレース範囲や実行時間測定範囲を特定するためのスタートアドレスやエンドアドレスは前記命令ブレイクアドレスレジスタに設定される。

【0072】SIO48は、マイクロコンピュータ10の外部に設けられたデバッグツール20との間でデータを送受信するためのものである。SIO48とデバッグツール20との間は、TXD/RXD(データ送受信ライン)で接続されている。

【0073】トレース情報出力部14とデバッグツール20との間は、DST[2:0]及びDPCOを出力するための4本のラインで接続されている。

【0074】トレース情報出力部14は、リアルタイムトレース機能を実現するためのものであり、図3で説明したCPUの命令実行ステータス情報(DST[2:0])及びPC絶対分岐が発生した際の分岐先のPC(プログラムカウンタ)値をトレース情報として専用4端子を介して外部に出力する。

【0075】以下DST[2:0]とDPCOを出力する構成について説明する。

【0076】CPU12はCPUの実行状態を表す信号(BR-ABS、BR-REL、IR-DONE)をライン72に出力する。BR-ABSはCPUがPC絶対分岐命令を実行した時に出力され、BR-RELはCPUがPC相対分岐命令を実行した時に出力され、IR-DONEはCPUが通常命令を実行した時に出力される。

【0077】トレース情報出力部14はライン72からCPUの実行状態を表す信号を受け取ると、当該信号に

基づきDST[1:0]に対応するステータスを出力する。

【0078】即ちIR-DONE信号を受け取った時は、CPUが通常命令を実行したのでDST[1:0]=00を出力する。BR-REL信号を受け取った時は、CPUがPC相対分岐命令を実行したのでDST[1:0]=01を出力する。BR-ABS信号を受け取った時は、CPUがPC絶対分岐命令を実行したのでDST[1:0]=10を出力する。DST[2]を出力する構成については後述する。このように、トレース情報出力部14がステータス情報出力手段として機能する。

【0079】また、トレース情報出力部14は、PC絶対分岐命令の実行(BR-ABS信号)に基づき、所定のタイミングでライン74を介して命令アドレス(飛び先のプログラムカウンタの値)を内部のシフトレジスタ52に取り込んで、BCLKに同期してシフトレジスタ52の値をDPCOからシリアルに出力する。

【0080】デバッグツール20は外部モニタ部61、トレース情報取得部22、実行時間測定部27を含み、パーソナルコンピュータ等により実現されるホストシステム66に接続される。

【0081】トレース情報取得部22は、3ビットのDST[2:0]と、分岐先のPC(プログラムカウンタ)値を表すDPCOを内部のトレース用メモリに格納する。トレース範囲が指定されている場合には、その範囲のDST[2:0]とDPCOをトレース用メモリに格納する。

【0082】実行時間測定部27は、指定された範囲の実行時間の測定を行う。

【0083】外部モニタ部61は、ホストシステム66から入力されるデバッグコマンドをプリミティブコマンドに変換(分解)するための処理を行う。そして、外部モニタ部61が、プリミティブコマンドの実行を指示するデータをオンチップモニタ部40に送信すると、オンチップモニタ部40が、指示されたプリミティブコマンドを実行するための処理を行うことになる。

【0084】従って、トレース範囲又は実行時間の測定範囲を指定したい場合、ユーザーはホストシステムからブレイクアドレス設定用のコマンドを入力する。このブレイクアドレス設定コマンドは、前記命令ブレイクアドレスレジスタへのライトコマンドに変換される。そして、オンチップモニタプログラムが、このプリミティブなライトコマンドを実行することで、命令ブレイクアドレスレジスタにトレース範囲又は実行時間の測定範囲のスタートアドレス及びエンドアドレスが設定される。即ち、オンチップモニタプログラム及び命令ブレイクアドレスレジスタが命令アドレス設定手段として機能する。

【0085】3. DST[2]を出力する構成
図7はDST[2]を出力するハードウェア構成の一例

である。

【0086】CPUとBCUはライン170、172、174、176、178で接続されて、信号のハンドシェイクを行っている。

【0087】図8は前記信号のタイムチャートの略図である。410はCPUクロックである。

【0088】CPU12はライン170を介してBCU26に命令のアリフェッチのリクエスト信号(CPU-IR-REQ)を送信し(図8の412参照)、ライン178を介してBCU26にフェッチすべき命令アドレス(IA)を送信する(図8の418参照)。CPU-IR-REQ412を受けたBCU26は当該信号を受け付けた旨の信号(CPU-IR-ACK)をライン172を介してCPU12に送信する(図8の414参照)。そして、前記命令アドレス(IA)から読み出した命令コード(IR-CODE)をライン176を介してCPUに送信し(図8の420参照)、命令コード送信した旨を示す信号(CPU-IR-VLD)をライン174を介してCPUに送信する(図8の416参照)。

【0089】ブレイクアドレスレジスタ0(180)、ブレイクアドレスレジスタ1(182)、ブレイクアドレスレジスタ2(184)はトレース範囲や実行時間測定範囲のスタートアドレス及びエンドアドレスを保持するためのレジスタである。本実施の形態では、前記オンチップモニタプログラムによりブレイクアドレスレジスタ2にエンドアドレスが、ブレイクアドレスレジスタ1にスタートアドレスが設定される。

【0090】比較器0(181)、比較器1(183)、比較器2(185)は、ライン198からくる命令アドレス(IA)と、前記ブレイクアドレスレジスタ0、ブレイクアドレスレジスタ1、ブレイクアドレスレジスタ2に保持されたアドレスとの比較を行う。

【0091】比較器1及び比較器2の比較結果は4段(8ビット)からなるFIFO186に入力される。FIFO186はライトポインタWR-PTR188に基づき比較結果が書き込まれ、リードポインタRD-PTR190に基づき比較結果を読み出され後段の検出信号制御回路204に出力される。

【0092】検出信号制御回路204は、フィリップフロップ回路206、微分回路208、EOR回路210を含む。このような回路構成により、ブレイクアドレスレジスタ2にマッチの比較結果を受けて検出信号202(DST[2]=1)を出力した後は、ブレイクアドレスレジスタ1にマッチするまで検出信号202(DST[2]=1)出力せず、ブレイクアドレスレジスタ1にマッチしたら検出信号202(DST[2]=1)出力するよう制御することができる。

【0093】次にFIFO186への比較結果の書き込みと読み出しのタイミングについて説明する。具体的には

ライトポインタWR-PTR188は、BCUが命令フェッチリクエストを受け付けた時（前記CPU-IR-ACK信号の出力されるタイミング）でインクリメントされ、インクリメントされたタイミングでライトポインタWR-PTR188の示す位置に前記比較結果を書き込む。

【0094】本実施の形態は前記前記CPU-IR-ACK信号の出力タイミングと比較器1（比較器2）からの比較結果を表す信号IR1-MATCH192（IR2-MATCH194）が出力されるタイミングが同期がとれるような構成を有しているからである（図8の422参照）。また、前記CPU-IR-ACK信号が出力されたということは、BCUがプリフェッチのリクエストを受け付けたことが確定しているから、分岐命令が実行されない限り確実に実行されるからである。

【0095】またリードポインタRD-PTRはライン196からCPU12が命令を実行した旨の信号（DAN信号）が出力されるタイミングでインクリメントされ、インクリメントされたタイミングでリードポインタRD-PTRの示す位置から前記比較結果を読み出す。

【0096】このようなタイミングで書き込み及び読み出しを行うのは、パイプライン処理に対応させるためである。即ち、FIFO186への書き込みの段階では命令がプリフェッチされた状態であり、CPU12が前記DAN信号を出力したときに実際の命令が終了するからである。

【0097】ここににおいて、CPU12が分岐命令を実行すると、分岐命令以前にプリフェッチされた命令は実行されない。このため、分岐命令以前にプリフェッチされた命令との比較結果はリセットする必要がある。従ってライン72を介して、CPU12からPC絶対分岐命令が実行された旨の信号（図6で説明したBR-ABS）でWR-PTR188とRD-PTR190はリセットされる。

【0098】図9（A）（B）は、分岐命令とリセットの関係を説明するための図である。図9（A）は分岐命令実行前のFIFOの状態を表しており、図9（B）はリセット後のFIFOの状態を表している。380にライトポインタがある時、分岐命令が実行されたとする。すると、プリフェッチ命令との比較結果382（斜線部分）は実行されない命令との比較結果となる。

【0099】従って、図9（B）に示すようにライトポインタWR-PTR、リードポインタRD-PTRも初期値にリセットされる。

【0100】4. デバッグツールの構成例

図10にデバッグツール20の構成例を示す。

【0101】CPU90は、ROM108に格納されるプログラムを実行したり、デバッグツール20の全体の制御を行うものである。送受信切替部92は、データの送信と受信とを切り替えるためのものである。クロック

制御部94は、CPU90のSCLK端子、アドレスアップカウンタ100、トレース用メモリ24に供給するクロックを制御するものである。このクロック制御部94には、マイクロコンピュータ10（SIO48）からのBCLKが入力される。クロック制御部94は周波数検出回路95、分周回路96を含む。周波数検出回路95は、BCLKが属する周波数範囲を検出して、その結果を制御レジスタ98に出力する。また分周回路96での分周比は制御レジスタ98により制御される。即ちCPU90により実行される外部モニタプログラム（外部モニタROM110に格納）が、制御レジスタ98からBCLKの周波数範囲を読み出す。そして、外部モニタプログラムは、この周波数範囲に応じた最適な分周比を決定し、この分周比を制御レジスタ98に書き込む。そして、分周回路96は、この分周比でBCLKを分周してSMC2を生成し、CPU90のSCLK端子に出力する。

【0102】アドレスアップカウンタ100は、トレース用メモリ24のアドレスをカウントアップするためのものである。セレクト102は、ライン122（アドレスアップカウンタ100が出力するアドレス）とライン124（アドレスバス120からのアドレス）のいずれかを選択し、トレース用メモリ24のアドレス端子にデータを出力する。マイクロコンピュータがデバッグモード時（オンチップデバッグプログラム実行時）にはライン124が選択され、ユーザーモード時（ユーザープログラム実行時）にはライン122が選択される。

【0103】またセレクト106は、ライン126（図6のトレース情報出力部14の出力であるDST[2:0]、DPCO）とライン128（データバス118）のいずれかを選択し、トレース用メモリ24のデータ端子にデータを出力したり、データ端子からデータを取り出す。

【0104】アドレスアップカウンタ100、トレース用メモリ24、セレクト102、セレクト106はトレース情報取得部22として機能する。

【0105】実行時間測定回路140は、分周回路142と実行時間保持用のレジスタ143～145を含み、実行時間測定部として機能する。分周回路142は、実時間を10μ、1μ、50nsにそれぞれ分周する。そしてライン126（図6のトレース情報出力部14の出力であるDST[2:0]、DPCO）からの情報に基づき、実行時間の測定の開始及び終了のタイミングを判断し、測定した実行時間の積算値を前記レジスタ143～145保持する。具体的にはDST[2]に奇数番目に1がきたら実行時間の測定を開始し、偶数番目に1がきたら測定を中断する動作を繰り返して、測定時間を積算する。

【0106】ROM108は外部モニタROM110（図6の外部モニタ部61に相当）を含み、外部モニタ

ROM110には、外部モニタプログラムが格納される。この外部モニタプログラムは、デバッグコマンドをプリミティブコマンドに変換するための処理を行う。RAM112は、CPU90のワーク領域となるものである。

【0107】RS232Cインターフェース114、パラレルインターフェース116は、図6のホストシステム66とのインターフェースとなるものであり、ホストシステム66からのデバッグコマンドはこれらのインターフェースを介してCPU90に入力されることになる。クロック生成部118は、CPU90を動作させるクロックなどを生成するものである。

【0108】次に本実施形態でのリアルタイムトレース処理について簡単に説明する。本実施形態では、図6のCPU12の命令実行のステータスを表す3ビットのDST[2:0]と、分岐先のPC値を表すDPCOをトレース用メモリ24に蓄える。そして、トレース用メモリ24に蓄えられたデータとユーザプログラムのソースコードとに基づいて逆アセンブルし、プログラムのトレースデータを合成する。このようにすることで、マイクロコンピュータ10とデバッグツール20との間の接続ラインの本数を少なくしながら、リアルタイムトレース機能を実現することが可能になる。

【0109】マイクロコンピュータがユーザモード時（ユーザプログラム実行時）においては、ライン122が選択され、セレクト102を介してアドレスアップカウンタ100の出力がトレース用メモリ24のアドレス端子に入力される。またマイクロコンピュータがユーザモード時には、ライン126が選択され、セレクト106を介してDST[2:0]、DPCOがトレースメモリ104のデータ端子に入力される。ここでアドレスアップカウンタ100には、まず最初に、データバス118、アドレスバス120を用いてCPU90により図11(A)に示すようなスタートアドレスが設定される。またアドレスアップカウンタ100のST/SP（開始/停止）端子には、トレース範囲を特定するDST[2]のラインが接続される。そして図11(B)に示すように、DST[2]のラインに第1のパルス390が入力されると、アドレスアップカウンタ100のアドレスアップカウントが開始する。そして、DST[2]のラインに第2のパルス392が入力されると、アドレスアップカウンタ100のアドレスアップカウントが停止し、トレース動作が停止する。このようにして、所望のトレース範囲でのデータ(DST[2:0]、DPCO)をトレース用メモリ24に蓄えることが可能になる。

【0110】一方、ユーザプログラム実行モードからデバッグモードに移行すると、ライン124が選択され、セレクト102を介してアドレスバス120からのアドレスがトレース用メモリ24のアドレス端子に入力され

る。またライン128が選択され、セレクト106を介してトレース用メモリ24からのデータがデータバス118に出力される。これにより、トレース用メモリ24に蓄えられたデータ(DST[2:0]、DPCO)を、デバッグモード時にCPU90（メインモニタプログラム）が読み出すことが可能になる。そして、読み出されたデータとユーザプログラムのソースコードとに基づいて、トレースデータを合成することが可能になる。

【0111】なお、トレース用メモリ24に蓄えられたデータをホストシステムに送信して、ホストシステムでトレース合成を行うようにしてもよい。

【0112】6. 電子機器

次に、以上の本実施形態のマイクロコンピュータを含む電子機器に関して説明する。

【0113】例えば図12(A)に電子機器の1つであるカーナビゲーションシステムの内部ブロック図を示し、図13(A)にその外観図を示す。カーナビゲーションシステムの操作はリモコン510を用いて行われ、GPSやジャイロからの情報に基づいて位置検出部520が車の位置を検出する。地図などの情報はCDROM530（情報記憶媒体）に格納されている。画像メモリ540は画像処理の際の作業領域になるメモリであり、生成された画像は画像出力部550を用いて運転者に表示される。マイクロコンピュータ500は、リモコン510、位置検出部520、CDROM530などのデータ入力源からデータを入力し、種々の処理を行い、処理後のデータを画像出力部550などの出力装置を用いて出力する。

【0114】図12(B)に電子機器の1つであるゲーム装置の内部ブロック図を示し、図13(B)にその外観図を示す。このゲーム装置では、ゲームコントローラ560からのプレーヤの操作情報、CDROM570からのゲームプログラム、ICカード580からのプレーヤ情報等に基づいて、画像メモリ590を作業領域としてゲーム画像やゲーム音を生成し、画像出力部610、音出力部600を用いて出力する。

【0115】図12(C)に電子機器の1つであるプリンタの内部ブロック図を示し、図13(C)にその外観図を示す。このプリンタでは、操作パネル620からの操作情報、コードメモリ630及びフォントメモリ640から文字情報に基づいて、ビットマップメモリ650を作業領域として、印刷画像を生成し、プリント出力部660を用いて出力する。またプリンタの状態やモードを表示パネル670を用いてユーザに伝える。

【0116】本実施形態のマイクロコンピュータ又はデバッグシステムによれば、図12(A)～図13(C)の電子機器を動作させるユーザプログラムの開発の容易化、開発期間の短縮化を図れるようになる。またマイクロコンピュータが実動作する環境と同じ環境で、ユーザ

プログラムのデバッグ作業を行うことができるため、電子機器の信頼性を高めることができる。また電子機器に組み込まれるマイクロコンピュータのハードウェアを小規模化、低コスト化できるため、電子機器の低コスト化も図れるようになる。更に少ない専用端子で指定された範囲のリアルタイムトレースや実行時間の測定を行うことができるためデバッグ効率の向上を図るとともに、マイクロコンピュータ上でユーザーが使用する端子を最大限に確保することが可能になる。

【0117】なお本実施形態のマイクロコンピュータを適用できる電子機器としては、上記以外にも例えば、携帯電話（セルラーフォン）、PHS、ページャ、オーディオ機器、電子手帳、電子卓上計算機、POS端末、タッチパネルを備えた装置、プロジェクタ、ワードプロセッサ、パーソナルコンピュータ、テレビ、ビューファインダ型又はモニタ直視型のビデオテープレコーダなど種々のものを考えることができる。

【0118】なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0119】例えば本発明の1本の端子で範囲指定を行える構成は本実施形態で説明したものが特に望ましいが、これに限定されるものではない。

【0120】また本実施の形態では指定されるブレイクポイントが2つの場合を例にとり説明したが、3つ以上ある場合でもよい。

【0121】またマイクロコンピュータやデバッグツールの構成も本実施形態で説明したものに限定されず、種々の変形実施が可能である。

【0122】

【図面の簡単な説明】

【図1】CPU置き換え型のICEの例を示す図である。

【図2】本実施形態の特徴について説明するための図である。

【図3】DST[2:0]の出力値とCPUの命令実行状態の関係を表した図である。

【図4】図4(A)(B)(C)は、本実施の形態における命令ブレイクにマッチした場合のDST[2:0]の出力を説明するための図である。

【図5】DST[2]とトレース範囲の関係を示す図である。

【図6】本実施形態のマイクロコンピュータ、デバッグシステムの構成例を示す機能ブロック図である。

【図7】DST[2]を出力するハードウェア構成の一例である。

【図8】CPUとBCU間の信号のタイムチャートの略図である。

【図9】図9(A)(B)は、分岐命令とFIFOのリセットの関係を説明するための図である。

【図10】デバッグツールの構成例を示す機能ブロック

図である。

【図11】図11(A)(B)はリアルタイムトレースについて説明するための図である。

【図12】図12(A)(B)(C)は、種々の電子機器の内部ブロック図の例である。

【図13】図13(A)(B)(C)は、種々の電子機器の外観図の例である。

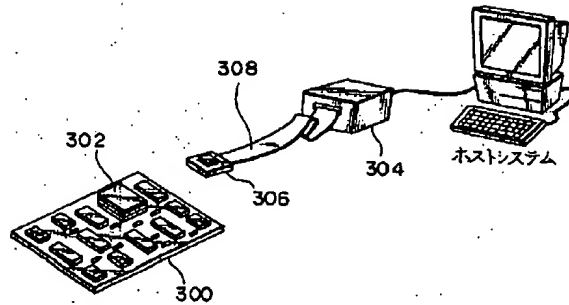
【符号の説明】

| | |
|-----|---------------------|
| 10 | マイクロコンピュータ |
| 12 | CPU(中央処理ユニット) |
| 13 | 内部レジスタ |
| 14 | トレース情報出力部 |
| 16 | 検出信号出力部 |
| 18 | 命令アドレス設定部 |
| 20 | デバッグツール |
| 22 | トレース情報取得部 |
| 24 | トレース用メモリ |
| 26 | BCU(バス制御ユニット) |
| 27 | 実行時間測定部 |
| 28 | 内部メモリ |
| 30 | クロック生成部 |
| 31 | バス |
| 32 | バス |
| 33 | 外部バス |
| 34 | 内部バス |
| 36 | 外部メモリ |
| 40 | オンチップモニタ部(第1のモニタ手段) |
| 42 | オンチップモニタROM |
| 44 | オンチップモニタRAM |
| 46 | 制御レジスタ |
| 48 | SIO |
| 52 | シフトレジスタ |
| 61 | 外部モニタ部(第2のモニタ手段) |
| 66 | ホストシステム |
| 90 | CPU |
| 92 | 送受信切替部 |
| 94 | クロック制御部 |
| 95 | 周波数検出部 |
| 96 | 分周回路 |
| 98 | 制御レジスタ |
| 100 | アドレスアップカウンタ |
| 102 | セレクタ |
| 106 | セレクタ |
| 108 | ROM |
| 110 | 外部モニタROM |
| 112 | RAM |
| 114 | RS232Cインターフェース |
| 116 | パラレルインターフェース |
| 118 | クロック生成部 |
| 140 | 実行時間測定回路 |

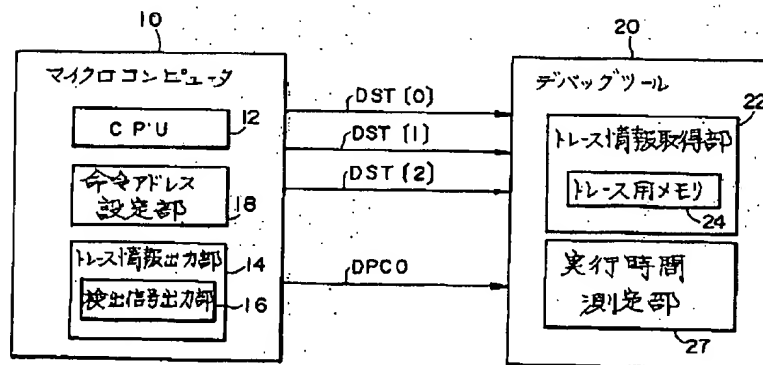
21
 142 分周回路
 143 レジスタ
 144 レジスタ
 145 レジスタ
 180 ブレークアドレスレジスタ0
 181 比較器0

22
 182 ブレークアドレスレジスタ1
 183 比較器1
 184 ブレークアドレスレジスタ2
 185 比較器2
 186 F I F O
 204 検出信号制御回路

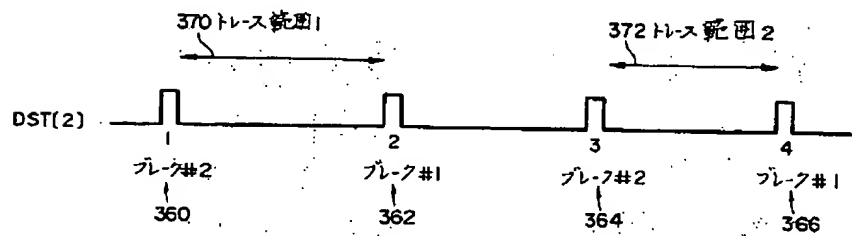
【図1】



【図2】



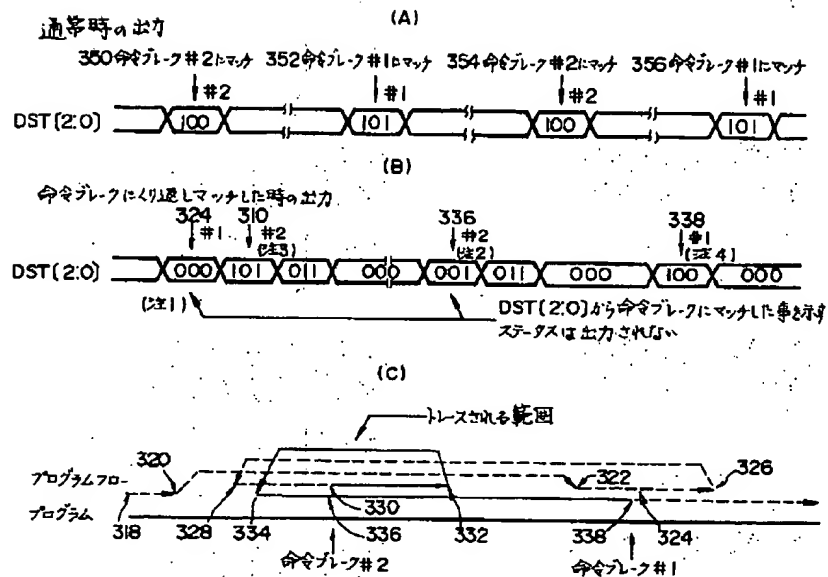
【図5】



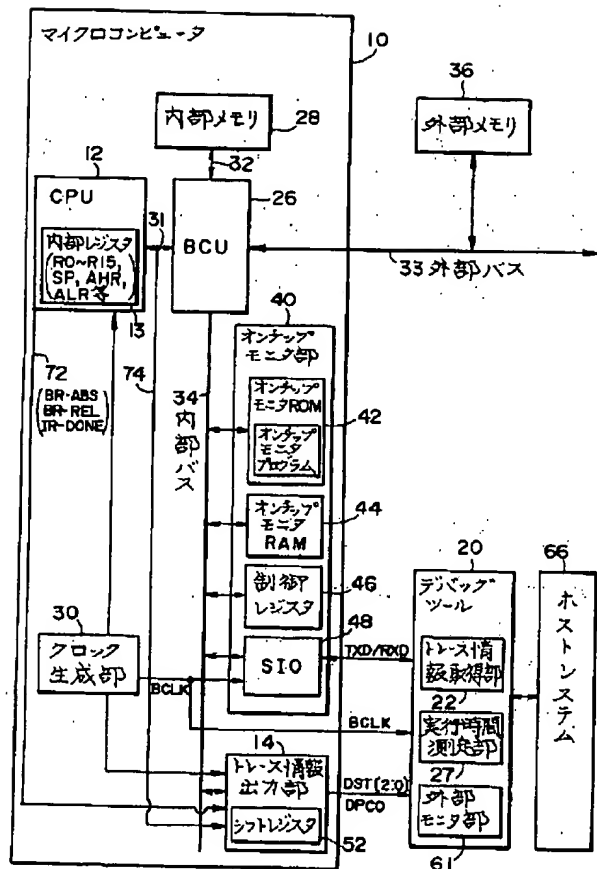
【図3】

| DST | | | CPUの命令実行状態 |
|-----|-----|-----|------------------------|
| (2) | (1) | (0) | |
| 0 | 0 | 0 | 通常命令実行終了(アドレス順に実行終了) |
| 0 | 0 | 1 | PC相対分岐命令実行 |
| 0 | 1 | 0 | PC絶対分岐命令実行 |
| 0 | 1 | 1 | アイドルサイクル |
| 310 | 1 | 0 | 通常命令実行時に命令ブレイクにマッチ |
| | 1 | 0 | PC相対分岐命令実行時に命令ブレイクにマッチ |
| | 1 | 1 | PC絶対分岐命令実行時に命令ブレイクにマッチ |
| 312 | 1 | 1 | ブレイク空間(デバッグモード) |

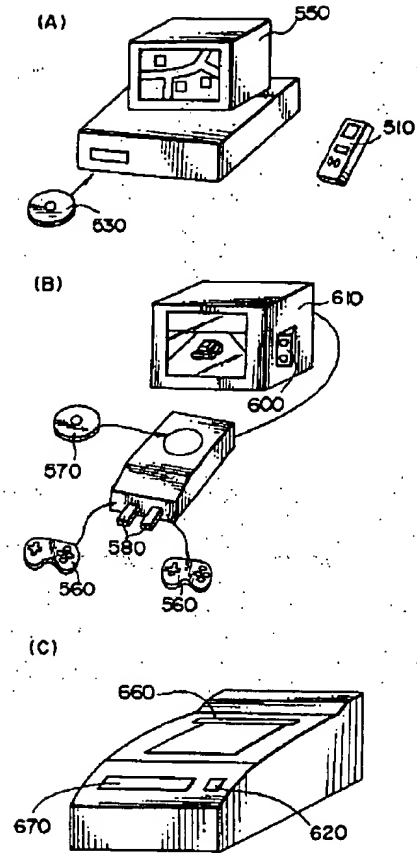
【図4】



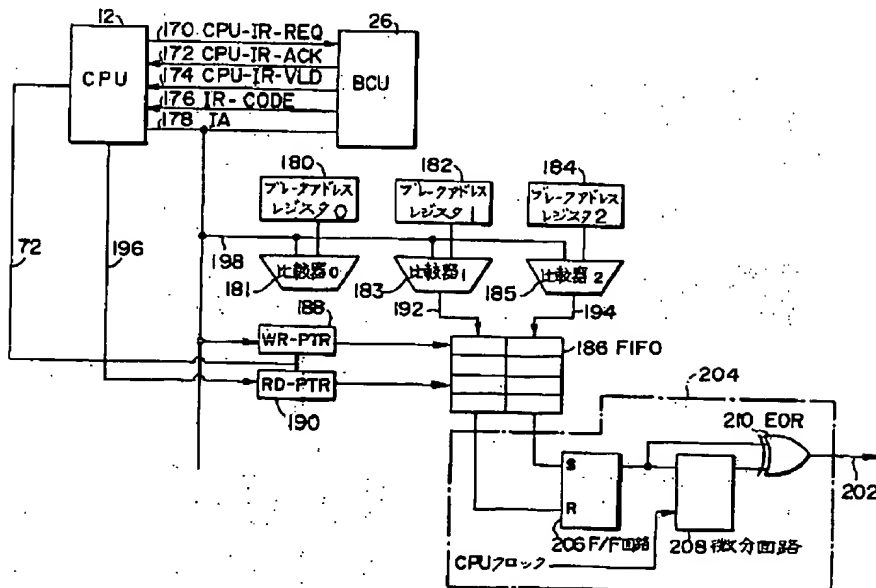
【図6】



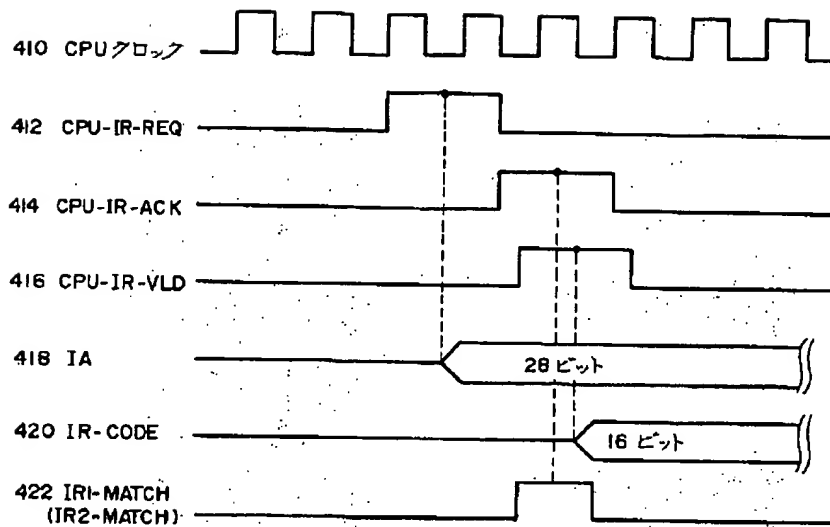
【図13】



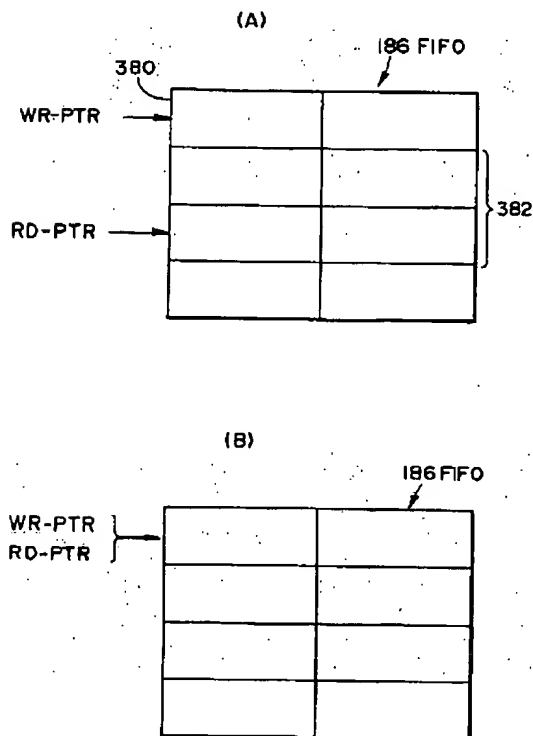
【図7】



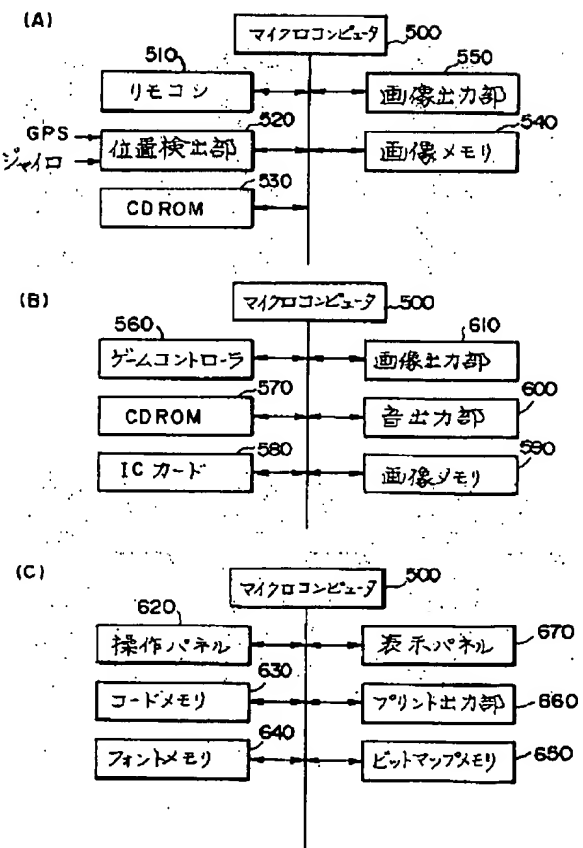
【図8】



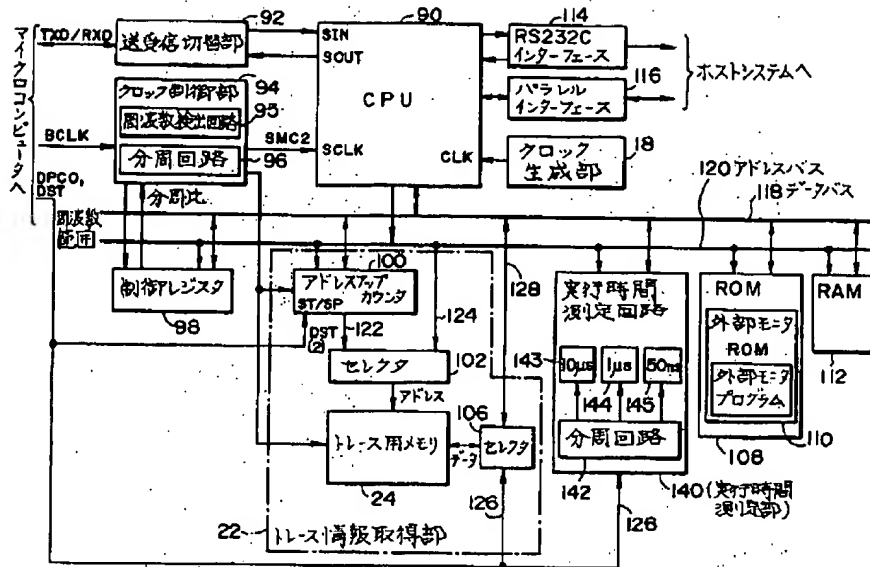
【図9】



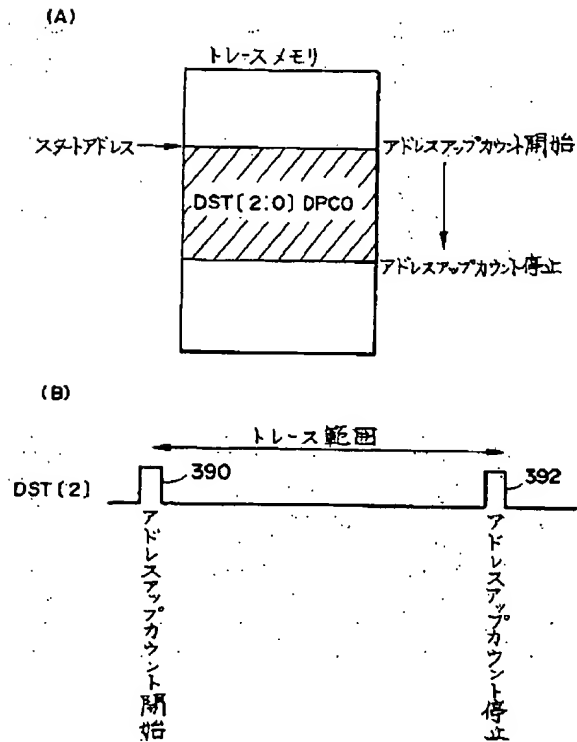
【図12】



【図10】



【図11】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.